David P. Stonehouse

University of Maryland University College

DATA 650 – Big Data Analytics

Fall 2016, Section 9040, Professor Gortcheva

Assignment 1: Text Mining Analysis Using R

## Part 1: Analyzing Course Descriptions

The first part of this assignment involved following a tutorial about textual analysis using R. The input data were 18 documents, each containing the description of a set of UMUC courses. The analysis objective was to investigate the patterns of usage of words across the descriptions and display the result in different ways. These include the document term matrix, a word cloud, a word frequency plot, a cluster graph, and a dendogram.

There are several data preparation steps in the process. Special characters, numbers and punctuation were removed to limit the analysis to just words. That is because our exercise is focused just on the relative use of words in the descriptions. Other textual analysis projects would include special characters, like parsing smily face sequences (e.g., colon followed by a right parenthesis) in tweets from a Twitter feed as a means of detecting sentiment (Microsoft, 2015). Special characters were removed using the R base gsub function for pattern replacement, while numbers and punctuation were removed using the removeNumbers and removePunctuation functions of the tm package.

Stop words are the common words in the language of the text being analyzed. In English, *a, the,* and *of* are examples of stop words. As Ganesan states, “if we remove the words that are very commonly used in a given language, we can focus on the important words instead”. Stop words were removed using the removeWords function of the tm package.

Stemming is done to remove suffixes like –ing and –ed so that two words that only differ by a suffix resolve to the same stemmed word. This ensures that the analysis focuses on the relationships between words as concepts. Typically, two words with the same base but two different suffixes are not related to different concepts. For this exercise, the stem words were removed using the stemDocument function of the SnowballC package.

Next, the processed text was transformed into a documents term matrix (DTM) using the DocumentTermMatrix function of the tm package. The matrix is *n* X *m*, for *n* documents and *m* terms, and each cell is the number of occurrences of the term for one of the documents. For the set of documents in the study, the sparsity was 90%, meaning that 90% of the cells had a frequency of zero. A DTM is useful to get some understanding about how words are commonly used. It is also useful as input for other analyses, like cluster analysis.

## Part 2: Textual Analysis of Raleigh City Council Meeting Minutes

## Introduction/Purpose

The student newspaper at Midtown University wants to monitor the work going on in Raleigh city government as a source for their “What’s going on in the city?” column. This is a semi-annual article highlighting projects going on in the city. The goal of the study is to see if by analyzing the minutes from city council meetings they will see the most important trends. To prove this idea out, they will study the minutes for the Raleigh City Council Growth and Natural Resources committee (City of Raleigh, 2016).

## Experimental Design / Method

***Talk about the doc source– more about the specific committee, how many documents, maybe sample of the template, why chosen***

The minutes for meetings of the Growth and Natural Resource Planning committee were downloaded from City of Raleigh (2016) from the City Council page. There were nine sets of meeting minutes, in MS Word format. The size of the documents ranged from 6000 to 34000 words. All of them were converted to text by opening them in MS Word and saving them to text. All of the documents follow a similar outline, almost all have a similar writing style and they use much of the same words. For example, all of the documents start by mentioning who was present.

Two main methods of textual analysis are a “bag of words” analysis, which handles terms individually, and semantic processing, which attempts to understand what a word is by the context. Similar to Part 1, the analysis for this study will only use “bag of words” techniques. However, approaches to extending this study with semantic processing techniques are discussed in the Future Research section.

Some of the techniques used in Part 1 are not going to work well for this analysis. A key problem is that there many words that are part of the meeting minutes “template” that do not give the user information about real trends in the data. Also, there are many “stop words” for this domain, like “zoning”, “councilor” and “develop”. These will be very common, but give no information about the specifics of topics being discussed. Finally, the secretary taking the minutes has their way of writing which causes certain words to be used a lot.

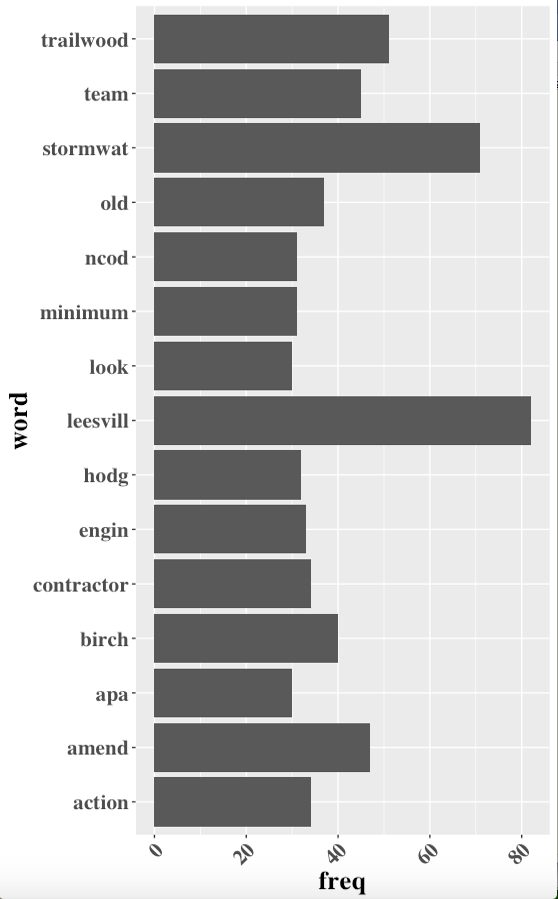
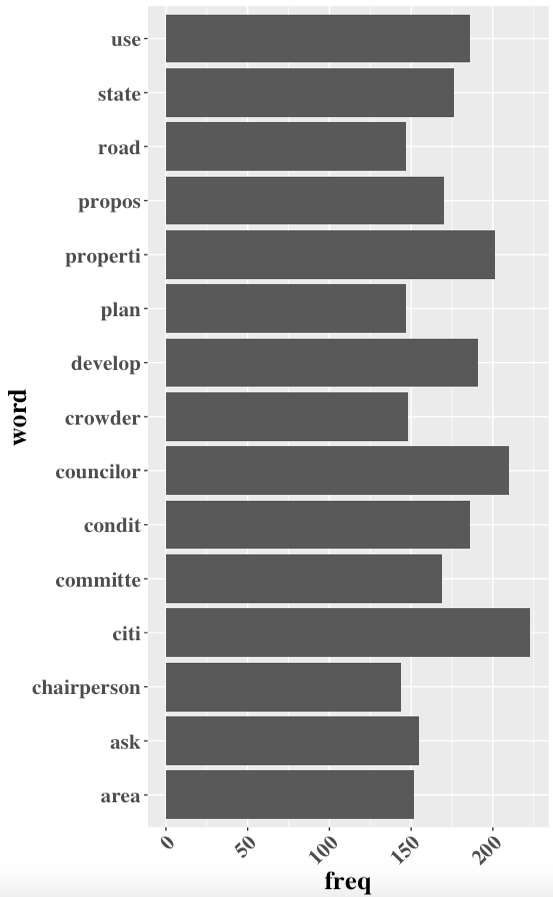
For all of those reasons, there is a need to do some additional transformation on the documents. The first is to increase our set of stop words by finding and removing words that are found in every single document. This was done using the *bounds* argument of the DocumentTermMatrix function. When the upper bound is set to 1 less than the number of documents, terms that are in every document are filtered out. Details of the effects of this approach will be shown in the Results section.

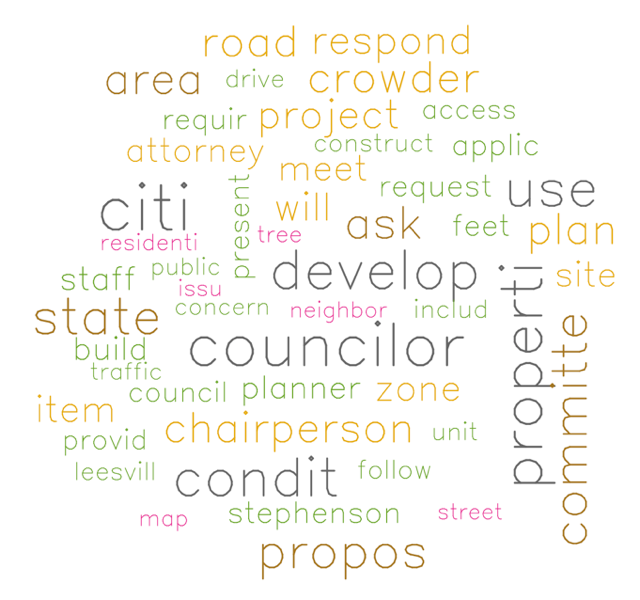
While increasing the stopwords can clean up the documents a little, but in these documents, there are many more stopwords. These stopwords are found in many but not all of the documents. The users want to focus on terms that are not so common, but are nonetheless important. An approach to giving visibility to such word is the term-frequency-inverse-document-frequency (tf-idf) measure (Manning, 2008). By using this approach, terms that are used a lot, but only in a few documents, are given a higher value than terms found many times, but in many documents. There are two steps in determining the tf-idf. First, the inverse document frequency (IDF) measure is determined for each term by log (N/df) where *N* is the total number of documents and *df* is the number of documents containing the term. Then each term in the document-term matrix are multiplied by the idf for the term. The sum of these values for each term is used to rank the importance of the terms. Again, the effects of using tf-idf will

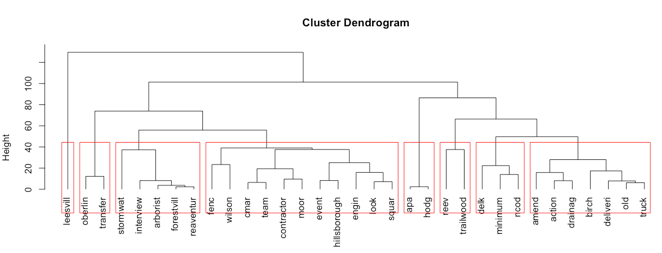
A key downside to both of these topics is that there could be a term related to a hot topic that is discussed in every single meeting. The users agreed that this would be okay to risk filtering those terms for the purposes of this study.

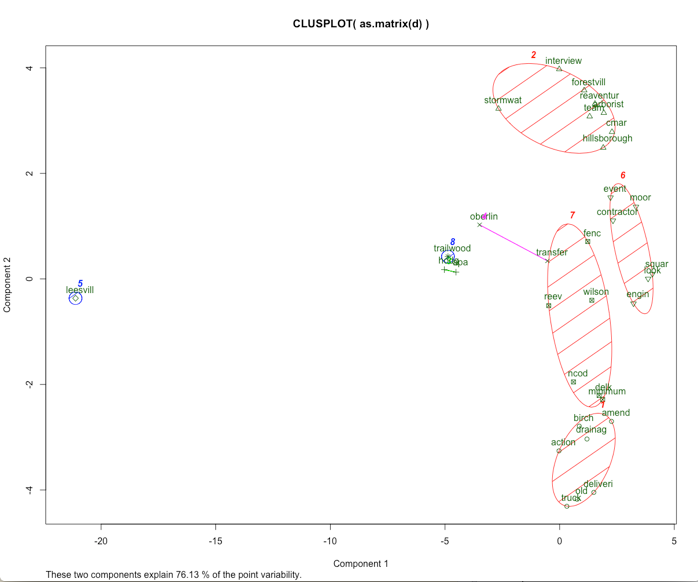
## Results

Illustrate findFreqTerms and word clouds









## Limitations and Future Research

There are many ways that this solution could be improved. Stem completion could make the results much easier to understand, but changes related to version 3.3.1 of R made it difficult to use.

Currently, the set of terms selected from the tf-idf matrix were chosen by playing with the threshold value for tf-idf. It would be better if the user could choose the “top N terms” instead.

The documents were converted to text just doing “Save as…” from MS Word. This process introduced a series of non-readable characters into the text. Either further substitution could be added, or possibly the process would be improved using the tm package’s readDOC function, which lets users read MS Word documents directly into a corpus.

The use of semantic processing would give users a lot of value-added information from these documents. One example is the use of a “neighborhood taxonomy”. The texts reference names of neighborhoods, many of which are made up of multiple words. It would be helpful if these were treated as a single term.

Similarly, street names could be identified through a taxonomy, or using analysis that analyzes nearby words. For example, when “Street” is in the minutes, textual analysis could attempt to create a term with the word or words before it. For example, “Old Leesville Road”, instead of the three terms “old” “Leesville” and “road”.

Further extending these items, any terms that have a geographic reference point (like roads or neighborhoods) could be displayed on a map with the size of the dot representing the relative importance of the term. Also, using a tool like Tableau, one could create an interactive map which shows highly correlated terms when clicking on or hovering over the dot.

## Conclusions

## References

Amazonaws.com. “Tutorial: Turning a Table into a Horizontal Bar Graph using ggplot2”. Retrieved from <http://rstudio-pubs-static.s3.amazonaws.com/4305_8df3611f69fa48c2ba6bbca9a8367895.html>

Ganesan, K., 2014. “All About Stop Words for Text Mining and Information Retrieval”. Retrieved from http://text-analytics101.rxnlp.com/2014/10/all-about-stop-words-for-text-mining.html

Greenacre, M., 2008. “Hierarchical cluster analysis”, retrieved from <http://www.econ.upf.edu/~michael/stanford/maeb7.pdf>

Manning, C. D.; Raghavan, P.; Schutze, H. (2008). "Scoring, term weighting, and the vector space model". Introduction to Information Retrieval (PDF). p. 100. doi:10.1017/CBO9780511809071.007. ISBN 9780511809071.

Microsoft (2015). “Text classification”. Retrieved from <https://gallery.cortanaintelligence.com/Experiment/Text-Classification-Step-1-of-5-data-preparation-3>

Raleigh, City of, 2016, City Council, retrieved from <https://www.raleighnc.gov/government/content/BoardsCommissions/Articles/CityCouncil.html>

Raleigh, City of, 2016, City of Raleigh Strategic Plan, retrieved from https://www.raleighnc.gov/government/content/BudgetManagement/Articles/StrategicPlan.html

Sonego, P (2011). “One R Tip A Day: Word Cloud in R”. Retrieved from http://onertipaday.blogspot.com/2011/07/word-cloud-in-r.html

Stack Overflow, 2015. “Text mining - Removing overly common words (occur in more than 80% of the documents)”. Retrieved from <http://stackoverflow.com/questions/25905144/removing-overly-common-words-occur-in-more-than-80-of-the-documents-in-r>

tfidf.com. “A Single-Page Tutorial - Information Retrieval and Text Mining”. Retrieved from http://www.tfidf.com/

## Appendix: Supporting Information